The NESSIE Project: Towards New Cryptographic Algorithms

Bart Preneel

Katholieke Univ. Leuven, Dept. Electrical Engineering-ESAT,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
bart.preneel@esat.kuleuven.ac.be

Abstract. In spite of more than 25 years of open research on cryptographic algorithms, many applications still use insecure cryptography. This paper attempts to explain this problem and tries to motivate why continuous research in this area is required. It also discusses the status of the NESSIE project. NESSIE (New European Schemes for Signature, Integrity and Encryption) is a 40-month research project (2000-2003) which intends to put forward a new generation of strong cryptographic algorithms, which have been obtained after an open call and an open evaluation process.

1 Introduction

Cryptographic algorithms play a crucial role in the information society. When we use our ATM or credit card, call someone on a mobile phone, get access to health care services, or buy something on the web, cryptographic algorithms are used to offer protection. These algorithms guarantee that nobody can steal money from our account, place a call at our expense, eavesdrop on our phone calls, or get unauthorized access to sensitive health data. It is clear that information technology will become increasingly pervasive: in the short term we expect to see more of e-government, e-voting, m-commerce, ...; beyond that we can expect the emergence of ubiquitous (or pervasive) computing, ambient intelligence,... These new environments and applications will present new security challenges, and there is no doubt that cryptographic algorithms and protocols will form part of the solution.

While cryptography is an essential component, the importance of cryptography should be put in the correct perspective. Indeed, failure of security systems can often be blamed on other reasons than failure of cryptography (see for example Anderson [1]). Nevertheless, cryptographic algorithms are part of the foundations of the security house, and any house with weak foundations will collapse. There is thus no excuse whatsoever to employ weak cryptography; nevertheless, we encounter weak cryptography more frequently than necessary and this for several reasons:

– Cryptography is a fascinating discipline, which tends to attract 'do-it-yourself' people, who are not aware of the scientific developments of the last 25 years;

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

their home-made algorithms can typically be broken in a few minutes by an expert;
- Use of short key lengths, in part due to export controls (mainly in the US, who dominates the software market) which limited key sizes to 40 bits (or 56 bits) for symmetric ciphers, 512 bits for factoring based systems (RSA) and discrete logarithm modulo a large prime (Diffie-Hellman). The US export restrictions have been lifted to a large extent in January and October 2000 (see Koops [15] for details). In several countries, domestic controls were imposed; the best known example is France, where the domestic controls were lifted in January 1999. Nevertheless, it can take a long time before all applications are upgraded.
- Progress in cryptanalysis: open academic research has started in the mid 1970ies; cryptology is now an established academic research discipline, and the IACR (International Association for Cryptologic Research) has more than 1000 members. As a consequence of this, increasingly sophisticated techniques are developed to break cryptosystems, but fortunately also to improve their security.
- Progress in computational power: Moore's law, which was formulated in 1965, predicts that every 18 months transistor density will double. Empirical observations have proved him right (at least for data density) and experts believe that this law will be holding for at least another 15 years. The variation of Moore's law for computational power states that the amount of computation that can be done for the same cost doubles every 18 month. This implies that a key for a symmetric algorithm will become thousand times cheaper to find after 15 years (or needs to increase in length by 10 bits to offer the same security). An even larger threat may be the emergence of new computer models: if quantum computers can be built, factoring may be very easy (a result by Shor of 1994 [27]). While early experiments are promising [30], experts are divided on the question whether sufficiently powerful quantum computers can be built in the next 15 years. For symmetric cryptography, quantum computers are less of a threat: they can reduce the time to search a $2n$-bit key to the time to search an n-bit key (using Grover's algorithm [11]). Hence doubling the key length offers an adequate protection.

As a consequence of all these observations, insecure cryptographic algorithms are much more common than they should be. In order to avoid these problems, adequate control mechanisms should be established at several levels:

- Substantial evaluation is necessary before an algorithm can be used; experts seem to agree that a period of 3 to 5 years is required between first publication and use of an algorithm.
- Continuous monitoring is required during the use of a primitive, to verify whether they are still adequate. Especially for public key primitives, which are parameterizable, a rigorous monitoring procedure is required to establish minimal key lengths.
- Adequate procedures should be foreseen to take an algorithm out of service or to upgrade an algorithm. Single DES is a typical example of an algorithm

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

which has been used beyond its lifetime (for most applications, 56 bits was no longer an adequate key length in the 1990ies); another example is the GSM encryption algorithm A5/1: experts agree that it is not as secure as believed (see e.g. Biryukov *et al.* [4]), but it is very difficult to upgrade it.

Especially the last problem should not be underestimated: for data authentication purposes, a new security weaknesses that is discovered will typically not influence older events, and long-term security can be achieved by techniques such as re-signing. However, for confidentiality the problem is much more dramatic: one cannot prevent that an opponent has access to ciphertext, and in certain cases (e.g., medical applications) secrecy for 50–100 years is required. This means that an encryption algorithm used now will need to withstand attacks employed in 2075. It is probably easier to imagine how hard it must have been to design in 1925 an encryption system that needed to be secure for 75 years. There is no reason to believe that this problem is easier at the beginning of the 21st century.

The remainder of this paper is organized as follows. Section 2 explains how the cryptographic challenge can be addressed, and what the role can be of integrated research projects. Section 3 introduces the NESSIE project and it approach. Section 4 discusses the status of the algorithms evaluated by NESSIE 6 months before the end of the project, and Sect. 5 presents some conclusions.

2 Addressing the Cryptography Challenge

Several elements are essential to tackle the cryptographic problem: standardization, research, and an open evaluation process.

A first element is the use of open standardization mechanisms, which are based on scientific evaluation rather than on commercial pressure. It is clear that algorithms should only be included in standards if they have received sufficient scrutiny. Moreover, the standardization body should establish adequate maintenance mechanisms in order to allow for timely revocation of algorithms or upgrade of parameters. One problem is that there are many standardization bodies, each with their own approach (EESSI, ETSI, IEEE, IETF, ISO, NIST, ANSI, ...). Algorithm revocation mechanisms are often too slow as standard maintenance procedures typically have large time constants. An example of a successful standardization effort is the NIST selection process for the Advanced Encryption Standard; this has been a 4-year effort resulting in the publication of FIPS 197 in November 2001 [9].

During the last 25 years, cryptographic research has been making substantial progress, and it is fair to say that cryptography has been evolving from an 'art' to a scientific discipline. Some of the most important developments have been made under influence of theoretical computer science: rigorous security definitions have been developed (which can takes sometimes many years to crystallize), and the reductionist approach has been introduced, also known as 'provable security.' This implies that formal proofs are provided that a weakness in a cryptographic primitive will imply that a hard problem can be solved. It should be noted however that this improves the state of the art significantly, but it does not solve

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

the crucial question: which problems are hard? Proving that a problem is hard is notoriously hard, or to quote James L. Massey *"A hard problem is a problem that nobody works on."* As a consequence, modern public-key cryptology depends on a limited set of problems believed to be hard. Most of these originate from algebraic number theory, the most popular ones are factoring the product of large primes and computing the discrete logarithm modulo a large prime. The discrete logarithm in other algebraic structures (defined by elliptic and hyper-elliptic curves) is also receiving attention. However, there is a clear need to perform more research on hard problems and to construct new schemes on other classes of hard problems. In the area of symmetric cryptology, a similar reductionist approach is being used. In this case however, the hard problems are typically not generic mathematical problems. The security is based on *'ad hoc'* designs, which have been developed based on years of experience and based on evaluation against existing and newly developed generic and specific attacks. In this area, performance is often very important. There is a need for new algorithms (stream ciphers, one-way functions, block ciphers) that have undergone a substantial security evaluation and that offer a better security performance trade-off for new environments (64-bit processors, smart cards, ultra-low power applications).

In order to guarantee the success of the standardization mechanisms, an independent and open evaluation effort is required to bridge the gap between the academic research community and the requirements of the applications. There are several reasons why such an effort is required:

- Academic research is more focused on providing a wide range of solutions with various properties rather than on providing a single solution.
- Academic research may not always fully specify all details of the algorithm, but rather focus on generic design approaches.
- Academic research often ignores certain 'small' problems that need to be addressed in applications and standards but that seem 'trivial.' However, such small details may have important security implications. A good example is the mechanism to indicate which hash function has been used together with the signature scheme, see Kaliski [12].
- Standardization bodies are not always in sink with the most recent academic developments.

Successful standards require a limited number of algorithms that are fully specified; this is required for interoperability. However, an algorithm is only useful if sufficient confidence has been built up, which illustrates the need for a thorough security evaluation. This may involve checking for statistical vulnerabilities and obvious weaknesses, applying known attacks, evaluation of the security against new attacks and a careful verification of all security proofs (the need for this can be illustrated by the error found by Shoup in the 7-year old OAEP security proof [28]; the error has been corrected for RSA-OAEP by Fujisaki *et al.* in [10]). The selection procedure for the algorithms requires careful benchmarking of security (on which problem is the primitive based, which model is required to prove security, how tight is the reduction,...), performance in various environments and other issues such as intellectual property. Typically standardization bodies

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

do not have the resources for such a careful benchmarking, and there is a clear need for an interface between the research community and the standardization bodies.

3 The NESSIE Project

NESSIE (New European Schemes for Signature, Integrity, and Encryption) is a research project within the Information Societies Technology (IST) Programme of the European Commission. The participants of the project are: Katholieke Universiteit Leuven (Belgium), coordinator, Ecole Normale Supérieure (France), Royal Holloway, University of London (U.K.), Siemens Aktiengesellschaft (Germany), Technion – Israel Institute of Technology (Israel), Université Catholique de Louvain (Belgium), and Universitetet i Bergen (Norway). NESSIE is a 40-month project, which started on January 1, 2000. Detailed and up to date information on the NESSIE project is available at http://cryptonessie.org/.

Next we discuss the NESSIE call and the NESSIE evaluation procedure which includes both a security and a performance evaluation; we also briefly discuss the software tools used during the evaluation.

3.1 The NESSIE Call

In the first year of the project, an open call for the submission of cryptographic algorithms, as well as for evaluation methodologies for these algorithms has been launched. The scope of this call has been defined together with the project industry board (PIB); the call was published in February 2000. The deadline for submissions was September 29, 2000. In response to this call NESSIE received 40 submissions, all of which met the submission requirements.

The NESSIE call includes a request for a broad set of algorithms providing date confidentiality, data authentication, and entity authentication. These algorithms include block ciphers, stream ciphers, hash functions, MAC algorithms, digital signature schemes, and public-key encryption and identification schemes (for definitions of these algorithms, see [18]). In addition, the NESSIE call asks for evaluation methodologies for these algorithms. While key establishment protocols are also very important, it was felt that they should be excluded from the call, as the scope of the call is already rather broad.

The scope of the NESSIE call is much wider than that of the AES call launched by NIST [20], which was restricted to 128-bit block ciphers. It is comparable to that of the RACE Project RIPE (Race Integrity Primitives Evaluation, 1988-1992) [26] (confidentiality algorithms were excluded from RIPE for political reasons) and that of the Japanese CRYPTREC project [6] (which also includes key establishment protocols and pseudo-random number generation). Another difference is that both AES and CRYPTREC intend to produce algorithms for government standards. The results of NESSIE will not be adopted by any government or by the European commission. However, the intention is that

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

relevant standardization bodies will adopt these results. As an example, algorithms for digital signature and hash functions may be included in the EESSI standardization documents which specify algorithms recommended for the European Electronic Signature Directive.

The call also specifies the main selection criteria which will be used to evaluate the proposals. These criteria are long-term security, market requirements, efficiency, and flexibility. Primitives can be targeted towards a specific environment (such as 8-bit smart cards or high-end 64-bit processors), but it is clearly an advantage to offer a wide flexibility of use. Security is put forward as the most important criterion, as security of a cryptographic algorithm is essential to achieve confidence and to build consensus.

For the *security requirements* of symmetric algorithms, two main security levels are specified, named *normal* and *high*. The minimal requirements for a symmetric algorithm to attain either the normal or high security level depend on the key length, internal memory, or output length of the algorithm. For block ciphers a third security level, *normal-legacy*, is specified, with a block size of 64 bits compared to 128 bits for the normal and high security level. The motivation for this request are applications such as UMTS/3GPP, which intend to use 64-bit block ciphers for the next 10-15 years. For the asymmetric algorithms, a varying security level is accepted, with as minimum about 2^{80} 3-DES encryptions.

If selected by NESSIE, the algorithm should preferably be available royalty-free. If this is not possible, then access should be non-discriminatory. The submitter should state the position concerning intellectual property and should update it when necessary.

The submission requirements are much less stringent than for AES, particularly in terms of the requirement for software implementations (only 'portable C' is mandatory).

3.2 The NESSIE Evaluation Procedure

As explained above, the NESSIE evaluation process takes into account the following elements: security, performance, and intellectual property status. About halfway into the project, a decision point has been inserted; at this stage, a subset of the submissions has been selected for a more detailed evaluation in the 2nd phase. Below we discuss the security and performance evaluation and the tools to support this evaluation.

Security Evaluation. A first step of the evaluation consists of basic checks on the submission, such as compliance with the call, working software, obvious weaknesses etc. The aim of this initial check was mainly to ensure that submissions were specified in a consistent and cogent form in time for the November 2000 workshop. It is vital for proper security assessments that the algorithms are fully and unambiguously described. This process required interaction with some submitters to ensure that the submissions were in the required form.

The next internal stage (November 2000) was to evaluate every submission in detail. An important principle adhered to during the evaluation is that if a

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

partner was involved in the design of a primitive, he should not be involved in the evaluation, and that all assessments are double-checked by a second project partner. If substantial flaws were discovered and confirmed, the evaluation of that primitive was stopped in order to optimize the project resources.

Next, an open workshop was organized in Egham (UK) on September 12-13, 2001 to discuss the security and performance analysis of the submissions. The presenters include both researchers from the NESSIE project, but also submitters, members from the NESSIE PIB, and members from the cryptographic community at large.

Following this workshop, a comprehensive security evaluation report has been published (D13 [19]). The document gives an overview of generic attacks on the different type of algorithms. Moreover, for each symmetric algorithm it presents a short description, the security claims by the designers, and the reported weaknesses and attacks. The part on asymmetric algorithms contains a discussion of security assumptions, security models, and of the methodology to evaluate the security. For each algorithm, a short description is followed by a discussion of the provable security (which security properties are proved under which assumptions) and of the concrete security reduction.

After the workshop, a subset of the algorithm has been selected (cf. Sect. 4.2 for more details). During the second half of the project, the remaining algorithms are being evaluated in more detail (see e.g., [23, 24]). The results of the 2nd phase have been presented at an open workshop on November 6-7, 2002 in Munich (Germany).

Performance Evaluation. Performance evaluation is an essential part in the assessment of a cryptographic algorithm. The candidates will be used on several platforms (PCs, smart cards, dedicated hardware) and for various applications. Some applications have tight timing constraints (e.g., payment applications, cellular phones); for other applications a high throughput is essential (e.g., high speed networking, hard disk encryption).

First a framework has been defined to compare the performance of algorithms on a fair and equal basis. It has been used for all evaluations of submitted candidates. First of all a theoretical approach has been established. Each algorithm is dissected into three parts: setup (independent of key and data), precomputations (independent of data, e.g., key schedule) and the algorithm itself (that must be repeated for every use). Next a set of four test platforms has been defined on which each candidate may be tested. These platforms are smart cards, 32-bit PCs, 64-bit machines, and Field Programmable Gate Arrays (FPGAs).

Then rules have been defined which specify how performance should be measured on these platforms. The implementation parameters depend on the platform, but may include RAM, speed, code size, chip area, and power consumption. On smart cards, only the following parameters will be taken into account, in decreasing order of importance: RAM usage, speed, code size. On PCs, RAM has very little impact, and speed is the main concern. On FPGAs, throughput, latency, chip area and power consumption will be considered. Unfortunately, the

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

limited resources of the project will not allow for the evaluation of dedicated hardware implementations (ASICs), but it may well be that teams outside the project can offer assistance for certain algorithms.

The project will consider the resistance of implementations to physical attacks such as timing attacks [13], fault analysis [3, 5], and power analysis [14]. For non constant-time algorithms (data or key dependence, asymmetry between encryption and decryption) the data or key dependence will be analyzed; other elements that will be taken into account include the difference between encryption and decryption, and between signature and verification operation. For symmetric algorithms, the key agility will also be considered.

This approach has resulted in the definition of a platform dependent test and in several platform dependent rekeying scenarios. Low-cost smart cards will only be used for block ciphers, MACs, hash functions, stream ciphers, pseudo-random number generation, and identification schemes.

In order to present performance information in a consistent way within the NESSIE project, a performance 'template' has been developed. The goal of this template is to collect intrinsic information related to the performance of the submitted candidates. A first part describes parameters such as word size, memory requirement, key size and code size. Next the basic operations are analyzed, such as shift/rotations, table look-ups, permutations, multiplications, additions, modular reduction, exponentiation, inversion,.... Then the nature and speed of precomputations (setup, key schedule, etc.) are described. Elements such as the dependence on the keys and on the inputs determine whether the code is constant-time or not. Alternative representations of the algorithms are explored when feasible.

Special software has been developed for automated performance testing on PCs and workstations. The status of the performance evaluation is presented in [25].

Tools. It is clear that modern computers and sophisticated software tools cannot replace human cryptanalysis. Nevertheless, software tools can play an important role in modern cryptanalysis. In most cases, the attacks found by the cryptanalyst require a large number of computational steps, hence the actual computation of the attack is performed on a computer. However, software and software tools can also be essential to find a successful way to attack a symmetric cryptographic algorithm; examples include differential and linear cryptanalysis, dependence tests, and statistical tests.

Within NESSIE, we distinguish two classes of tools. The general tools are not specific for the algorithms to be analyzed. Special tools, which are specific for the analysis of one algorithm, are implemented when, in the course of the cryptanalysis of an algorithm, the need for such a tool turns up.

For the evaluation of the symmetric submissions, a comprehensive set of general tools is available within the project. These tools are in part based on an improved version of the tools developed by the RIPE (RACE Integrity Primitives Evaluation) project [26]. These test include more than 20 statistical tests.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

The NESSIE project is also developing a new generic tool to analyze block ciphers with differential [2] and linear cryptanalysis [17]. This tool is based on a general description language for block ciphers.

The software for these tools will not be made available outside the project, but all the results obtained using these tools will be made public in full detail.

4 The Algorithms Evaluated by NESSIE

4.1 The NESSIE Submissions

The cryptographic community has responded very enthusiastically to the call. Thirty nine algorithms have been received, as well as one proposal for a testing methodology. After an interaction process, which took about one month, all submissions comply with the requirements of the call. There are 26 symmetric algorithms:

- seventeen block ciphers, which is probably not a surprise given the increased attention to block cipher design and evaluation as a consequence of the AES competition organized by NIST. They are divided as follows:
  - six 64-bit block ciphers: CS-Cipher, Hierocrypt-L1, IDEA, Khazad, MISTY1, and Nimbus;
  - seven 128-bit block ciphers: Anubis, Camellia, Grand Cru, Hierocrypt-3, Noekeon, Q, and SC2000 (none of these seven come from the AES process);
  - one 160-bit block cipher: Shacal; and
  - three block ciphers with a variable block length: NUSH (64, 128, and 256 bits), RC6 (at least 128 bits), and SAFER++ (64 and 128 bits).
- six synchronous stream ciphers: BMGL, Leviathan, LILI-128, SNOW, SOBER-t16, and SOBER-t32.
- two MAC algorithms: Two-Track-MAC and UMAC; and
- one collision-resistant hash function: Whirlpool.

Thirteen asymmetric algorithms have been submitted:

- five asymmetric encryption schemes: ACE Encrypt, ECIES, EPOC, PSEC, and RSA-OAEP (both EPOC and PSEC have three variants);
- seven digital signature algorithms: ACE Sign, ECDSA, ESIGN, FLASH, QUARTZ, RSA-PSS, and SFLASH; and
- one identification scheme: GPS.

Approximately[1] seventeen submissions originated within Europe (6 from France, 4 from Belgium, 3 from Switzerland, 2 from Sweden), nine in North America (7 USA, 2 from Canada), nine in Asia (8 from Japan), three in Australia and three in South America (Brazil). The majority of submissions originated within industry (27); seven came from academia, and six are the result of

[1] Fractional numbers have been used to take into account algorithms with submitters over several continents/countries – the totals here are approximations by integers, hence they do not add up to 40.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

a joint effort between industry and academia. Note however that the submitter of the algorithm may not be the inventor, hence the share of academic research is probably underestimated by these numbers.

All submissions are available on the NESSIE web site [19].

4.2 Selection for the 2nd Phase

On September 24, 2001, the NESSIE project has announced the selection of candidates for the 2nd phase of the project. Central to the decision process has been the project goal, that is, to come up with a portfolio of strong cryptographic algorithms. Moreover, there was also a consensus that every algorithm in this portfolio should have a unique competitive advantage that is relevant to an application.

It is thus clear that an algorithm could not be selected if it failed to meet the security level required in the call. A second element could be that the algorithm failed to meet a security claim made by the designer. A third reason to eliminate an algorithm could be that a similar algorithm exists with better security (for comparable performance) or with significantly better performance (for comparable security). In retrospect, very few algorithms were eliminated because of performance reasons; this can be motivated in part because the large number of submissions did not allow for an in-depth performance evaluation during the 1st phase. It should also be noted that the selection was more competitive in the area of block ciphers, where many strong contenders were considered. The motivation for the decisions is given in [22]. Note that the submissions originating from industry performed best, while only one submission from academia only made it the 2nd phase.

Designers of submitted algorithms were allowed to make small alterations to their algorithms; the main criterion to accept these alterations is that they should improve the algorithm and not substantially invalidate the existing security analysis. More information on the alterations can be found on the NESSIE webpages [19].

The selected algorithms are listed below; altered algorithms are indicated with a *. Block ciphers:

- IDEA: MediaCrypt AG, Switzerland;
- Khazad*: Scopus Tecnologia S.A., Brazil and K.U.Leuven, Belgium;
- MISTY1: Mitsubishi Electric Corp., Japan;
- SAFER++64, SAFE++128: Cylink Corp., USA, ETH Zurich, Switzerland, National Academy of Sciences, Armenia;
- Camellia: Nippon Telegraph and Telephone Corp., Japan and Mitsubishi Electric, Japan;
- RC6: RSA Laboratories Europe, Sweden and RSA Laboratories, USA;
- Shacal: Gemplus, France.

Here IDEA, Khazad, MISTY1 and SAFER++64 are 64-bit block ciphers. Camellia, SAFER++128 and RC6 are 128-bit block ciphers, which will be compared to AES/Rijndael [7, 9]. Shacal is a 160-bit block cipher based on SHA-1 [8]. A

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

256-bit version of Shacal based on SHA-256 [21] has also been introduced in the second phase; this algorithm will be compared to an RC-6 and a Rijndael [7] variant with a block length of 256 bits (note that this variant is not included in the AES standard). The motivation for this choice is that certain applications (such as the stream cipher BMGL and certain hash functions) can benefit from a secure 256-bit block cipher.
Synchronous stream ciphers:

- SOBER-t16, SOBER-t32: Qualcomm International, Australia;
- SNOW*: Lund Univ., Sweden;
- BMGL*: Royal Institute of Technology, Stockholm and Ericsson Research, Sweden.

It has become clear in the Spring of 2002 that SOBER-t16, SOBER-t32, and SNOW have security flaws which imply that they do not meet the stringent security requirements imposed by NESSIE. Moreover, BMGL submission is rather slow (more than 10 times slower than AES), hence while is useful as a pseudo-random bit generator, it is not suitable as a high speed stream cipher for bulk data.
MAC algorithms and hash functions:

- Two-Track-MAC: K.U.Leuven, Belgium and debis AG, Germany;
- UMAC: Intel Corp., USA, Univ. of Nevada at Reno, USA, IBM Research Laboratory, USA, Technion, Israel, and Univ. of California at Davis, USA;
- Whirlpool*: Scopus Tecnologia S.A., Brazil and K.U.Leuven, Belgium.

The hash function Whirlpool will be compared to the new FIPS proposals SHA-256, SHA-384 and SHA-512 [21].
Public-key encryption algorithms:

- ACE-KEM*: IBM Zurich Research Laboratory, Switzerland (derived from ACE Encrypt);
- EPOC-2*: Nippon Telegraph and Telephone Corp., Japan;
- PSEC-KEM*: Nippon Telegraph and Telephone Corp., Japan (derived from PSEC-2);
- ECIES*: Certicom Corp., USA and Certicom Corp., Canada
- RSA-OAEP*: RSA Laboratories Europe, Sweden and RSA Laboratories, USA.

Digital signature algorithms:

- ECDSA: Certicom Corp., USA and Certicom Corp., Canada;
- ESIGN*: Nippon Telegraph and Telephone Corp., Japan;
- RSA-PSS: RSA Laboratories Europe, Sweden and RSA Laboratories, USA;
- SFLASH*: BULL CP8, France;
- QUARTZ*: BULL CP8, France.

Identification scheme:

- GPS*: Ecole Normale Supérieure, Paris, BULL CP8, France Télécom and La Poste, France.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

Many of the asymmetric algorithms have been updated at the beginning of phase 2. For the asymmetric encryption schemes, these changes were driven in part by the recent cryptanalytic developments, which occurred after the NESSIE submission deadline [10, 16, 28]. A second reason for these changes is the progress of standardization within ISO/IEC JTC1/SC27 [29]. The standards seem to evolve towards defining a hybrid encryption scheme, consisting of two components: a KEM (Key Encapsulation Mechanism), where the asymmetric encryption is used to encrypt a symmetric key, and a DEM (Data Encapsulation Mechanism), which protects both secrecy and integrity of the bulk data with symmetric techniques (a "digital envelope"). This approach is slightly more complicated for the encryption of a short plaintext, but it offers a more general solution with clear advantages. Three of the five NESSIE algorithms (ACE Encrypt, ECIES and PSEC-2) have been modified to take into account this development. At the same time some other improvements have been introduced; as an example, ACE-KEM can be based on any abstract group, which was not the case for the original submission ACE Encrypt. Other submitters decided not to alter their submissions at this stage. For further details, the reader is referred to the extensive ISO/IEC draft document authored by V. Shoup [29]. The NESSIE project will closely monitor these developments. Depending on the progress, variants such as RSA-KEM defined in [29] may be studied by the NESSIE project.

For the digital signature schemes, there seem to be less problems with security-related issues; nevertheless, three out of five schemes (ESIGN, QUARTZ and SFLASH) have been altered. In this case, there are particular reasons for each algorithm (correction for the security proof to apply, improve performance, or preclude a new attack). The other two have not been modified. It should also be noted that PSS-R, which offers very small storage overhead for the signature, has not been submitted to NESSIE. QUARTZ offers very short signatures (16 bytes), but the signing algorithm is very slow and the public key is large.

4.3 Intellectual Property

While it would be ideal for users of the NESSIE results that all algorithms recommended by NESSIE were in the public domain, it is clear that this is for the time being not realistic. The users in the NESSIE PIB have clearly stated that they prefer to see royalty-free algorithms, preferably combined with open source implementations. However, providers of intellectual property typically have different views.

One observation is that in the past, there has always been a very large difference between symmetric and asymmetric cryptographic algorithms. Therefore it is not so surprising that NIST was able to require that the designers of the block cipher selected for the AES would give away all their rights, if their algorithm was selected; it is clear that this is not a realistic expectation for the NESSIE project.

We will attempt to summarize the intellectual property statements of the submissions retained for the 2nd phase. Note however that this interpretation is only indicative; for the final answer the reader is referred to the intellectual

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

property statement on the NESSIE web page [19], and to the submitters themselves.

Eleven out of 22 algorithms that are still being considered are in the public domain, or the submitters indicate that a royalty-free license will be given. These are the block ciphers Khazad, Misty1, Shacal, Safer++, the stream cipher BMGL, the MAC algorithms Two-Track-MAC and UMAC, the hash function Whirlpool, the public-key encryption algorithm RSA-OAEP[2], and the digital signature schemes SFLASH and RSA-PSS.[2]

Royalty-free licenses will be given under very broad conditions for the block cipher Camellia, for the public-key encryption algorithms EPOC-2 and PSEC-KEM, and for the digital signature scheme ESIGN (the main exception occurs when someone holds IPR on a primitive in the same class and implements the complete class of recommended algorithms).

The block cipher IDEA is free for non-commercial use only; for commercial applications a license is required. Licenses under fair, reasonable and non-discriminatory terms will be given for the public-key encryption scheme ACE-KEM (the detailed license conditions are rather complex) and for the digital signature scheme QUARTZ.

Additions to the 'reasonable and non-discriminatory' terms are required for the public-key algorithms ECDSA and ECIES and the identification scheme GPS; it is required that the license holder reciprocates some of his rights.

Finally, the submitters of RC6 have indicated that for the time being they do not longer support RC6 for new applications due to IPR problems.

It is clear that intellectual property is always a complex issue, and it will not be possible to resolve this completely within the framework of NESSIE. However, IPR issues may play an important role in the final selection process.

5 Conclusions

Open evaluation initiatives such as AES, RIPE, CRYPTREC, and NESSIE can bring a clear benefit to the cryptographic research community and to the users and implementors of cryptographic algorithms. By asking cryptographers to design concrete and fully specified schemes, they are forced to make choices, to think about performance optimizations, and to consider all the practical implications of their research. While leaving many options and variants in a construction may be very desirable in a research paper, it is often confusing for a practitioner. Implementors and users can clearly benefit from the availability of a set of well defined algorithms, that are described in a standardized way.

The developments in the last years have also shown that this approach can result in a better understanding of the security of cryptographic algorithms. We have also learned that concrete security proofs are an essential tool to build confidence, particularly for public key cryptography (where constructions can be reduced to mathematical problems believed to be hard) and for constructions

[2] This statement does not hold for the variants of RSA with more than two primes.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

that reduce the security of a scheme in an efficient way to other cryptographic algorithms. At the same time, we have learned from the many alterations made to the asymmetric primitives that this field may not be completely mature and that it is essential to study proofs for their correctness.

Another conclusion from the NESSIE project is that there is a clear need for a very fast and highly secure stream cipher (as the submitted candidates do not seem to satisfy the requirements).

The NESSIE project is inviting the community at large to further analyze the candidates for the 2nd phase, and to offer comments on their security, performance and intellectual property status. The project is accepting comments until mid December 2002, and the final selection will probably be announced by February 2003.

Acknowledgments. I would like to thank all the members of and the contributors to the NESSIE project. The work described in this paper has been supported by the Commission of the European Communities through the IST Programme under Contract IST–1999–12324.

References

1. R.J. Anderson, "Why cryptosystems fail," *Communications ACM*, Vol. 37, No. 11, November 1994, pp. 32–40.
2. E. Biham, A. Shamir, *"Differential Cryptanalysis of the Data Encryption Standard,"* Springer-Verlag, 1993.
3. E. Biham, A. Shamir, "Differential fault analysis of secret key cryptosystems," *Advances in Cryptology, Proceedings Crypto'97, LNCS 1294*, B. Kaliski, Ed., Springer-Verlag, 1997, pp. 513–525.
4. A. Biryukov, A. Shamir, D. Wagner, "Real time cryptanalysis of A5/1 on a PC," *Fast Software Encryption, LNCS 1978*, B. Schneier, Ed., Springer-Verlag, 2000, pp. 1–18.
5. D. Boneh, R. A. DeMillo, R. J. Lipton, "On the importance of checking cryptographic protocols for faults," *Advances in Cryptology, Proceedings Eurocrypt'97, LNCS 1233*, W. Fumy, Ed., Springer-Verlag, 1997, pp. 37–51.
6. CRYPTREC project, http://www.ipa.gov.jp/security/enc/CRYPTREC/index-e.html.
7. J. Daemen, V. Rijmen, "AES proposal Rijndael," September 3, 1999, available from http://www.nist.gov/aes.
8. FIPS 180-1, *"Secure Hash Standard,"* Federal Information Processing Standard (FIPS), Publication 180-1, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., April 17, 1995.
9. FIPS 197 *"Advanced Encryption Standard (AES),"* Federal Information Processing Standard (FIPS), Publication 197, National Institute of Standards and Technology, US Department of Commerce, Washington D.C., November 26, 2001.
10. E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern, "RSA-OAEP is secure under the RSA assumption," *Advances in Cryptology, Proceedings Crypto'01, LNCS 2139*, J. Kilian, Ed., Springer-Verlag, 2001, pp. 260–274.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

11. L.K. Grover, "A fast quantum mechanical algorithm for database search," *Proc. 28th Annual ACM Symposium on Theory of Computing,* 1996, pp. 212–219.
12. B. Kaliski, "On hash function firewalls in signature schemes," *Topics in Cryptology, CT-RSA 2002, LNCS 2271*, B. Preneel, Ed., Springer-Verlag, 2002, pp. 1–16.
13. P. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," *Advances in Cryptology, Proceedings Crypto'96, LNCS 1109*, N. Koblitz, Ed., Springer-Verlag, 1996, pp. 104–113.
14. P. Kocher, J. Jaffe, B. Jun, "Differential power analysis," *Advances in Cryptology, Proceedings Crypto'99, LNCS 1666*, M.J. Wiener, Ed., Springer-Verlag, 1999, pp. 388–397.
15. B.-J. Koops, "Crypto law survey," http://rechten.kub.nl/koops/cryptolaw.
16. J. Manger, "A chosen ciphertext attack on RSA Optimal Asymmetric Encryption Padding (OAEP) as standardized in PKCS #1 v2.0," *Advances in Cryptology, Proceedings Crypto'01, LNCS 2139*, J. Kilian, Ed., Springer-Verlag, 2001, pp. 230–238.
17. M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," *Advances in Cryptology, Proceedings Crypto'94, LNCS 839*, Y. Desmedt, Ed., Springer-Verlag, 1994, pp. 1–11.
18. A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *"Handbook of Applied Cryptography,"* CRC Press, 1997.
19. NESSIE, http://www.cryptonessie.org.
20. NIST, AES Initiative, http://www.nist.gov/aes.
21. NIST, *"SHA-256, SHA-384, SHA-512,"* Washington D.C.: NIST, US Department of Commerce, Draft, 2000.
22. B. Preneel, B. Van Rompay, L. Granboulan, G. Martinet, S. Murphy, R. Shipsey, J. White, M. Dichtl, P. Serf, M. Schafheutle, E. Biham, O. Dunkelman, V. Furman, M. Ciet, J.-J. Quisquater, F. Sica, L. Knudsen, H. Raddum, *"NESSIE Phase I: Selection of Primitives"* NESSIE Report, September 2001, available from [19].
23. B. Preneel, S.B. Örs, A. Biryukov, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, P. Serf, E. Biham, E. Barkan, O. Dunkelman, M. Ciet, F. Sica, L.R. Knudsen, H. Raddum *"Update on the selection of algorithms for further investigation during the second round,"* NESSIE Deliverable D18, March 2002, available from [19].
24. B. Preneel, A. Biryukov, E. Oswald, B. Van Rompay, L. Granboulan, E. Dottax, S. Murphy, A. Dent, J. White, M. Dichtl, S. Pyka, M. Schafheutle, P. Serf, E. Biham, E. Barkan, O. Dunkelman, M. Ciet, F. Sica, L. Knudsen, M. Parker, H. Raddum, *"NESSIE Security Report,"* NESSIE Deliverable D20, October 2002, available from [19].
25. B. Preneel, B. Van Rompay, B. Van Rompay, S.B. Örs, A. Biryukov, L. Granboulan, E. Dottax, M. Dichtl, M. Schafheutle, P. Serf, S. Pyka, E. Biham, O. Dunkelman, J. Stolin, M. Ciet, J-J. Quisquater, F. Sica, H. Raddum, M. Parker, *"Performance of Optimized Implementations of the NESSIE Primitives,"* NESSIE Deliverable D21, October 2002, available from [19].
26. RIPE, *"Integrity Primitives for Secure Information Systems. Final Report of RACE Integrity Primitives Evaluation (RIPE-RACE 1040)," LNCS 1007*, A. Bosselaers, B. Preneel, Eds., Springer-Verlag, 1995.
27. P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proc. 35nd Annual Symposium on Foundations of Computer Science*, S. Goldwasser, Ed., IEEE Computer Society Press, 1994, pp. 124–134.
28. V. Shoup, "OAEP reconsidered," *Advances in Cryptology, Proceedings Crypto'01, LNCS 2139*, J. Kilian, Ed., Springer-Verlag, 2001, pp. 239–259.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.

29. V. Shoup, *"A Proposal for an ISO Standard for Public Key Encryption,"* Version 2.0, September 17, 2001, available from http://www.shoup.net.
30. L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, ''Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature*, 414, 2001, pp. 883–887.

Appeared in *Information Security Applications, 3rd International Workshop, WISA 2002*, Lecture Notes in Computer Science, Springer-Verlag, pp. 33–16, 2002.